

Повышение эффективности тензорной аппроксимации для сжатия изображений с помощью заранее обученного словаря

М.К. Чобану, Д.В. Макаров

Национальный исследовательский университет «МЭИ» (НИУ «МЭИ»)

makarovdmv@gmail.com

Аннотация — Методы тензорной декомпозиции все чаще применяются в цифровой обработке сигналов для улучшения характеристик стандартных алгоритмов. В данной работе рассматривается применение одного из наиболее эффективных методов тензорной декомпозиции. Показана возможность и перспективы его применения в качестве метода сжатия изображений. Также предложен подход с применением словаря, состоящего из заранее вычисленных элементов, позволяющий повысить эффективность сжатия изображений.

Ключевые слова — тензор, сжатие с потерями, аппроксимации цепочкой тензоров, сигнално-зависимые фильтры, разреженное представление.

I. ВВЕДЕНИЕ

Тензорный анализ и теория тензорных аппроксимаций играют все более важную роль в области вычислительной математики и численного анализа. Эффективное представление d -мерного тензора (массива с d индексами) небольшим числом параметров может дать возможность работать с данными размерности d , равной 10, 100 или даже 1000 (такие проблемы возникают в квантовой молекулярной динамике, финансовом моделировании, при решении стохастических уравнений в частных производных).

Предлагается альтернативный метод сжатия изображений на основе тензорной аппроксимации. В условиях экспоненциального роста объемов передачи, хранения и обработки визуальной информации [1] применение данного подхода является оправданным.

Применённые методы тензорной аппроксимации показали хорошие результаты при работе с изображениями в оттенках серого, кроме того существует большой потенциал для дальнейшего усовершенствования методов.

II. ТЕНЗОРНАЯ АППРОКСИМАЦИЯ

Идея аппроксимации тензоров заключается в нахождении закономерностей среди элементов тензора и приближении исходного тензора декомпозицией тензоров меньшей размерности. Описание методов аппроксимации и современных алгоритмов их реализации дано в [2].

Известными методами аппроксимации являются:

- 1) Каноническая аппроксимация (наиболее удачный алгоритм получения такой декомпозиции это CANDECOMP/PARAFAC [3]).
- 2) Декомпозиция Такера (реализации N-mode PCA и N-mode SVD [4]).
- 3) Аппроксимация цепочкой тензоров (Tensor-Train Decomposition [5], ТТ).

Каноническая аппроксимация наиболее эффективна и позволяет представить тензор \mathbf{T} любой размерности в виде набора двумерных тензоров $(\mathbf{U}_1, \mathbf{U}_2, \dots, \mathbf{U}_d)$, с помощью которых можно вычислить элементы \mathbf{T} как сумму:

$$\mathbf{T}(i_1, i_2, \dots, i_d) = \sum_{\alpha=1}^r \mathbf{U}_1(i_1, \alpha) \cdot \mathbf{U}_2(i_2, \alpha) \cdot \dots \cdot \mathbf{U}_d(i_d, \alpha),$$

где $\mathbf{T}(i_1, i_2, i_3, \dots, i_d)$ – исходный тензор размера $n_1 \times n_2 \times n_3 \times \dots \times n_d$, \mathbf{U}_k – двумерный тензор размера $n_k \times r$, r – канонический ранг.

Таким образом, возможна аппроксимация тензора \mathbf{T} (при $n_1=n_2=\dots=n_d=n$) числом элементов, оцениваемым как $O(d \cdot n \cdot r)$. Преимуществом метода является то, что если ранг r невелик, то тензор можно представить очень компактно. Однако алгоритмы получения канонической декомпозиции не являются стабильными, и даже если известно, что существует декомпозиция малого ранга r_{min} , то нет гарантии, что алгоритму удастся получить аппроксимацию с таким рангом.

Этого недостатка лишен метод аппроксимации цепочкой тензоров ТТ (Tensor-Train Decomposition). По числу элементов в аппроксимации метод приближается к канонической аппроксимации, при этом существует стабильный алгоритм для получения такой аппроксимации.

Идея метода заключается в представлении тензора большой размерности цепочкой тензоров размерности 3:

$$\mathbf{T}(i_1, i_2, \dots, i_d) = \sum_{\alpha_2, \dots, \alpha_d} \mathbf{G}_1(\alpha_0, i_1, \alpha_1) \cdot \dots \cdot \mathbf{G}_d(\alpha_{d-1}, i_d, \alpha_d), \quad (1)$$

где \mathbf{G}_k – тензор размера $r_{k-1} \times n_k \times r_k$, $\alpha_0 = \alpha_1 = 1$. При этом аппроксимация выполняется с точностью ε :

$$\|\mathbf{T} - \mathbf{T}'\| \leq \varepsilon \|\mathbf{T}\|,$$

где \mathbf{T} – исходный тензор, \mathbf{T}' – тензор, которым был аппроксимирован \mathbf{T} . Тензоры \mathbf{G}_k вычисляются с помощью алгоритма SVD (Singular Value Decomposition), что гарантирует получение декомпозиции для любых данных.

Сумму (1) можно представить матричным произведением, т.к. каждому значению i_k в трехмерном массиве \mathbf{G}_k соответствуют матрицы \mathbf{H}_k :

$$\mathbf{T}(i_1, i_2, \dots, i_d) = \mathbf{H}_1(i_1) \cdot \mathbf{H}_2(i_2) \cdot \dots \cdot \mathbf{H}_d(i_d), \quad (2)$$

где \mathbf{H}_k – матрица размером $r_{k-1} \times r_k$. При этом необходимо, чтобы $r_0 = r_d = 1$ (результатом матричного произведения должен быть единственный элемент). Т.к. ранги r_k обычно невелики, то по числу элементов аппроксимация приближается к канонической и ее размер оценивается как $O((d-2) \cdot n \cdot r^2 + 2 \cdot n \cdot r)$. Алгоритм метода и численные результаты изложены в [5].

III. ПРИМЕНЕНИЕ ВЕЙВЛЕТНОЙ ЦЕПОЧКИ ФИЛЬТРОВ (WAVELET TENSOR-TRAIN)

Сжатие с помощью аппроксимации осуществляется за счет выигрыша по числу элементов. Но снижение числа элементов не всегда приводит к сжатию, т.к. можно получить меньше элементов, но они при этом будут иметь более сложную структуру, чем исходные, вследствие чего будут закодированы менее эффективно. Такая ситуация может произойти в методе ТТ: можно получить меньшее число элементов в аппроксимации, но они будут более сложными (дробные числа со знаком, кодируемые 8-ю байтами), чем исходные (целые без знака, кодируются 1 байтом). Поэтому возможно, что в результате сжатие будет невелико.

Применение вейвлетной цепочки фильтров (Wavelet Tensor-Train, WTT [6]) является модификацией метода ТТ, позволяющей представить исходный сигнал в более разреженном виде. Идея заключается в использовании тензоров \mathbf{H}_k в качестве фильтров для исходного сигнала (т.е. \mathbf{H}_k используется как матрица без преобразования в 3-х мерный тензор). Т.к. фильтры получены на основе сигнала (сигнально-зависимые), то они позволяют эффективно представить исходный сигнал в разреженном виде (т.е. с большим количеством нулевых элементов, [7]). Такой массив коэффициентов имеет малую энтропию и хорошо сжимается. В результате фильтры являются сигнально-зависимыми, поэтому необходимо хранить их вместе со сжатым изображением.

Чтобы фильтры не были очень большими, необходимо ограничить их ранг, иначе они будут в точности представлять сигнал и их будет неудобно хранить. Введем параметр r_{max} , задающий максимальный ранг фильтров. Алгоритм получения фильтров приведен в [6].

Применение фильтров к изображению сводится к последовательному перемножению фильтра и матрицы изображения.

Сигнал можно восстановить, т.к. фильтры являются ортогональными (свойство алгоритма SVD) т.е. выполняется соотношение

$$\mathbf{H}_k \cdot \mathbf{H}_k^T = \mathbf{H}_k^T \cdot \mathbf{H}_k = \mathbf{E}_{r_k},$$

где \mathbf{H}_k – ортогональная матрица фильтра (размера $r_k \times r_k$), \mathbf{H}_k^T – транспонированная матрица \mathbf{H}_k , \mathbf{E}_{r_k} – единичная матрица размера $r_k \times r_k$.

A. Сравнение эффективности ТТ и WTT

Для сравнения эффективности методов WTT и ТТ оценим число бит на пиксель (*bit per pixel, bpp*), выраженное через энтропию результата:

$$bpp = \frac{entropy(result) \cdot sizeof(result)}{n \cdot m},$$

где *result* – массив, содержащий выходной результат работы ТТ/WTT, *entropy* – функция, вычисляющая энтропию элементов массива, *sizeof* – определяет количество элементов в массиве, *n, m* – размеры матрицы изображения. Результатом работы WTT является разреженный массив коэффициентов и фильтры. Максимальный ранг фильтров (r_{max}) в тесте равен 3. Число бит на пиксель будем сравнивать на соответствующих значениях качества восстановленных изображений, полученных с помощью показателя *PSNR* (Peak signal-to-noise ratio, пиковое отношение сигнала к шуму).

Результат сравнения представлен на рис. 1. В качестве входных данных использовалось стандартное тестовое изображение «Lena» в оттенках серого. Для оценки энтропии результирующие значения были приведены к 16-ти битным целым.

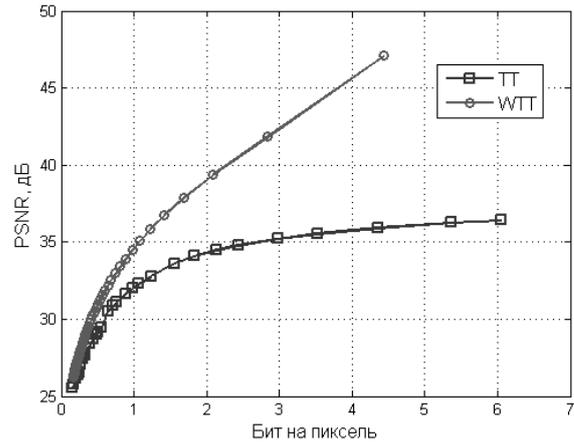


Рис. 1. Сравнение числа бит на пиксель, оцененного с помощью энтропии, для методов WTT и ТТ

Из графика видно, что WTT позволит эффективнее сжать изображение, чем ТТ.

B. Применение WTT для сжатия изображений

Возможность использования предвычисленных фильтров уже была рассмотрена в работе [8], где было установлено, что уменьшение максимального ранга фильтров r_{max} , до 4 и менее позволяет сделать фильтры

универсальными (практически независимыми от сигнала). Такой подход позволяет избежать хранения фильтров вместе со сжатым изображением, но, вместе с тем, снижает эффективность фильтров. Результат применения такого подхода представлен на рис. 2. Эффективность сравнима с JPEG (по качеству и скорости).

IV. ИСПОЛЬЗОВАНИЕ WTT С ЗАРАНЕЕ ВЫЧИСЛЕННЫМ СЛОВАРЕМ

Возможен подход с применением WTT не для всего изображения, а для блока 8×8 , аналогично применению дискретного косинусного преобразования (ДКП) в JPEG. Результат исследования показал, что WTT работает эффективнее ДКП, но при этом не удается сделать фильтры сигнально независимыми за счет уменьшения ранга фильтров r_{max} , как в [8]. Т.е. для каждого блока необходим уникальный фильтр, что сводит на нет все преимущества от применения WTT.

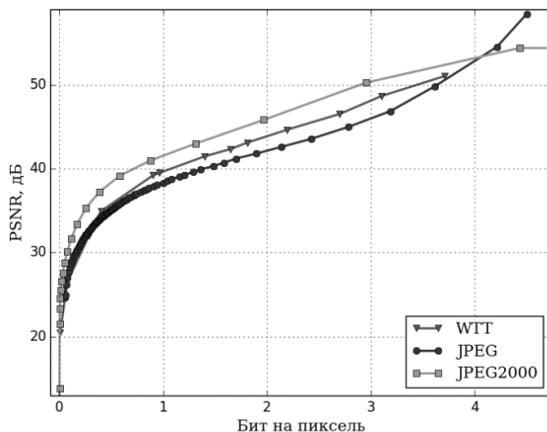


Рис. 2. Сравнение WTT с JPEG2000 и JPEG для изображения «Lena»

В тоже время была проанализирована эффективность применения одного фильтра, вычисленного для блока 8×8 , на других блоках, отличных от исходного. В результате было выявлено, что фильтр может работать эффективно не только на исходном блоке, а на целом классе блоков [9]. В таблице 1 представлена статистика применения нескольких фильтров на наборе 8192 блоков. Для квантования ДКП-коэффициентов использовалась матрица квантования Q из стандарта JPEG [10]. Сравнение производилось на основе энтропии квантованных данных при фиксированном значении $PSNR$.

Из таблицы видно, что, например, фильтр 4006 может работать эффективнее ДКП более чем на 30 % на 635 блоках из 8192. Это дает возможность вычислить набор (словарь) фильтров, в котором каждый фильтр будет соответствовать определенному классу блоков.

Статистика применения WTT-фильтров для блока размером 8×8

Номер фильтра	Кол-во блоков, на которых фильтр работает как минимум не хуже ДКП	Кол-во блоков, на которых фильтр работает как минимум на 30% лучше ДКП	Кол-во блоков, на которых фильтр работает хуже ДКП
3823	2967	691	4534
4006	3147	635	4510
3264	3061	604	4527
3947	2694	602	4896
5821	2770	596	4826

Кроме того, следует отметить, что фильтр WTT полученный для конкретного блока будет эффективен не только для этого блока, но и для блоков схожих с исходным блоком. На рис. 3 отражен этот факт. На графике представлена степень сжатия блока с помощью ДКП и WTT-фильтра в зависимости от искажений этого блока. Величина случайного искажения определяет наибольшее случайное число, которое прибавлено к каждой компоненте блока 8×8 . Кривая с номером 1 соответствует энтропии после применения к блоку ДКП и затем матрицы квантования Q . Следующая кривая(2) соответствует применению WTT-фильтра, вычисленного для исходного блока, а энтропия оценивается при фиксированном качестве ($PSNR$), соответствующем значению $PSNR$ после применения ДКП. Кривая 3 аналогична кривой 2 с тем отличием, что WTT-фильтр вычислялся для каждого искаженного блока.

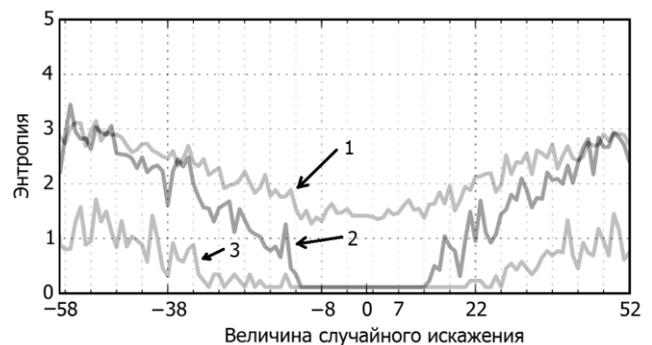


Рис. 3. Сравнение эффективности WTT-фильтра на блоках схожих с исходным

Процедура построения словаря заключается в поиске таких фильтров, которые наиболее эффективны и работают на наибольшем количестве блоков для обучения. Более подробно процедура обучения была изложена в работе [9], в которой также была приведена оценка эффективности применения такого словаря в сравнении с ДКП (рис. 4). На рисунке отражено сравнение степени сжатия нескольких изображений, одно из которых входило в набор изображений на которых обучался словарь, остальные нет.

А. Способы выбора элементов из словаря

При использовании WTT со словарем наиболее сложным вопросом является процедура выбора элемента словаря в зависимости от входного блока.

Для этого можно предложить несколько способов:

1) Перебор фильтров из словаря. Наиболее простой и очевидный способ. Он использовался в [9] для построения графика рис. 4. Преимуществом метода является его безошибочность. Он точно может выбрать наилучший фильтр для входного блока, либо определить, что такого фильтра нет в словаре. Недостатком метода является длительность работы. Хотя скорость восстановления изображения по-прежнему остается быстрой (сравнима с JPEG).

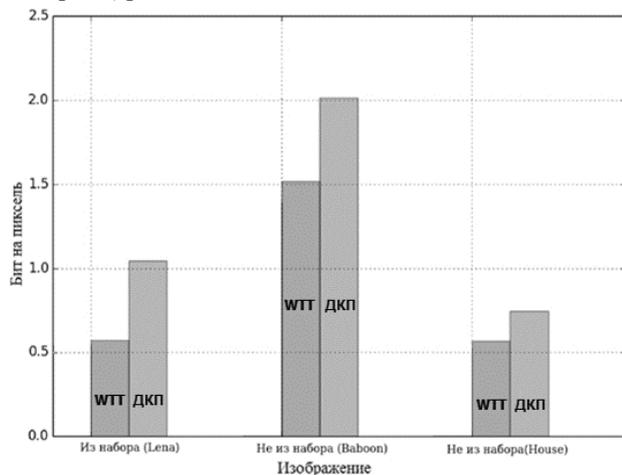


Рис. 4. Сравнение эффективности WTT со словаря и ДКП

2) Возможно применение метода с использованием хеш-функций. В этом случае в словаре хранятся не только фильтры, но и хеш-значения блоков на которых тот или иной фильтр эффективно работает. Значением хеш-функции является число, отражающее какую-либо «особенности» блока. Если блоки имеют одинаковое хеш-значение, то с большой вероятностью эти блоки идентичны. Кроме того, хеш-функция должна быть выбрана таким образом, чтобы хеш-значения «близких» блоков отличались незначительно. Простым примером такой хеш-функции может является функция:

$$h(\mathbf{B}) = \sum_{i=1}^8 \sum_{j=1}^8 b_{ij}, \quad (3)$$

где \mathbf{B} – матрица входного блока 8×8 , b_{ij} – элемент матрицы \mathbf{B} . Т.к. хеш является числом, то можно в словаре хранить отсортированный массив таких чисел, что позволит свести сложность операции поиска фильтра к бинарному поиску $O(\log N)$. Естественно, что функция (3) будет совпадать для многих блоков (коллизия хеш-функции), поэтому можно использовать не одну функцию, а несколько. Главным преимуществом такого подхода является скорость работы. К недостатку метода можно отнести возможность ошибки из-за коллизии хеш-функций и, как показала практика, при

увеличении размера словаря сильно возрастает и вероятность ошибки выбора фильтра. Так же в отличие от первого метода нет возможности выбрать фильтр, для блока, который не входил в набор для обучения и не является «схожим» с каким-либо блоком из набора для обучения.

3) С помощью обучения нейронной сети. Безусловно между блоками, для которых эффективен какой-либо фильтр WTT существует закономерность. Возможно при обучении нейронная сеть сможет выявить эту закономерность и объединить в себе преимущества первого и второго методов.

V. ЗАКЛЮЧЕНИЕ

Применение WTT позволяет достичь эффективности сжатия изображений сравнимой с алгоритмом JPEG. Кроме того, существует возможность применения заранее обученного словаря, что позволяет увеличить эффективность сжатия блоков изображения в сравнении с ДКП.

Представленный подход можно использовать как дополнение или альтернативу ДКП, чтобы повысить эффективности современных алгоритмов сжатия изображений и видео. В случае дополнения, ДКП применяется к блоку, когда эффективный фильтр для блока отсутствует в словаре.

Для метода сжатия с применением WTT и словаря по-прежнему остаются открытыми вопросы:

- 1) Выбор оптимального размера блока для метода WTT со словарем. В работе использовались блоки 8×8 для удобства сравнения с ДКП. Но, возможно, эффективнее будет использование блоков меньшего размера.
- 2) Поиск наиболее эффективного метода выбора фильтров из словаря для сжатия входного блока.
- 3) Сжатие для цветных изображений.

ЛИТЕРАТУРА

- [1] Дворкович В., Чобану М. Проблемы и перспективы развития систем кодирования динамических изображений // MediaVision. 2011. № 2. С. 55-64.
- [2] Kolda T. G., Bader B. W. Tensor decompositions and applications // SIAMRev. 2009. № 51. P. 455–500.
- [3] Kiers H. A. L. A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity // Journal of Chemometrics. 1998. № 12. P. 171-255.
- [4] Vasilescu M. A. O., Terzopoulos D., Multilinear analysis of image ensembles: Tensor-Faces, in ECCV 2002 // Proceedings of the 7th European Conference on Computer Vision. 2002. V. 2350 of Lecture Notes in Computer Science. P. 447-460.
- [5] Oseledets I.V. Tensor-train decomposition // SIAM J. Sci.Comput. 2011. V. 33. №. 5. P. 2295–2317.
- [6] Oseledets I.V. Algebraic wavelet transform via quantics tensor train decomposition // INM RAS. 2010. Preprint 2010-03.
- [7] Чобану М. К. Многомерные многоскоростные системы обработки сигналов. М.: Техносфера, 2009. 480 с.

- [8] Чобану М.К., Макаров Д.В. Сжатие изображений с помощью тензорной аппроксимации // Проблемы разработки перспективных микро- и наноэлектронных систем - 2014. Сборник трудов / под общ. ред. академика РАН А.Л. Стемпковского. М.: ИПИМ РАН, 2014. Часть IV. С. 109-112.
- [9] Чобану М.К., Макаров Д.В. Сжатие изображений с помощью тензорной аппроксимации и заранее вычисленного словаря // Цифровая обработка сигналов и ее применение – DSPA 2015. – М.: Издательский дом Медиа Паблишер, 2015 г. – том 2. – 711-715 стр.
- [10] ISO/IEC 10918-1:1992 Informatontotechnology – Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines. ISO/IEC JTC1/SC29/WG10, 1992.

Improving the efficiency of the tensor approximation for image compression by using the trained dictionary

M.K. Tchobanou, D.V. Makarov

National Research University “MPEI”, makarovdmv@gmail.com

Keywords — tensor, tensor decomposition, lossy image compression, train tensor approximation, wavelet tensor-train, signal-dependent filters, sparse representation.

ABSTRACT

Nowadays tensor decomposition methods are frequently used in digital signal processing to improve characteristics of standard algorithms. In this paper, we consider the use of one of the most effective methods of tensor decomposition. We show that there is possibility to use this method of tensor decomposition for image compression. In addition, we introduce an approach, which uses a dictionary with pre-calculated elements that increases the efficiency of image compression.

Tensor analysis and the theory of tensor approximations are playing an increasingly important role in computational mathematics and numerical analysis. An efficient representation of a tensor (by tensor we mean only an array with d indices) with a small number of parameters may give us ability to work with d -dimensional problems, with d being as high as 10, 100, or even 1000. The application of this approach is warranted when there is an exponential growth of transmission, storage and processing of visual information [1].

The purpose of tensor approximation is to find regularities among elements of an input tensor and decompose it into tensors with smaller dimension and smaller total count of elements [2].

There are three effective methods of tensor approximation:

- Canonical decomposition (the CANDECOMP, PARAFAC algorithms [3]).
- Tucker decomposition (the N-mode PCA and N-mode SVD algorithms [4]).
- Tensor-Train Decomposition (the TT decomposition algorithm [5]).

There is an effective way to represent d -dimensional tensors by using the canonical decomposition. It decomposes initial tensor in set of two-dimensional tensors.

Thus, it is possible to approximate a tensor of size n by n by n by ... by n with a number of elements, which are measured as $O(d \cdot n \cdot r)$. The advantage of the method is that if the rank r is small, the tensor can be represented very compactly. But the computation of the canonical rank is an NP-hard problem and numerical algorithms for computing an approximate representation in such cases might fail.

The Tucker decomposition is stable but has exponential in d number of parameters. It is suitable for “small” dimensions, especially for the three-dimensional case. For large d it is not suitable.

The TT decomposition does not have these drawbacks. There is a stable algorithm for calculating TT decomposition and the number of elements in the approximation is close to canonical approximation.

Sparse representation (like wavelet [6]) is an efficient way for image compression. In this case, we obtain projection of an image to more suitable basis where the image will have a large number of repeating elements. It allows encoding the image more effectively.

The Wavelet Tensor-Train (WTT, [7]) is a modification of TT decomposition method for computing a sparse representation. This is possible to compute a set of orthogonal matrices, which is a good basis for the image. Then matrices will be used as filters for the image. Using filters for sparse image computation leads to sequence of matrix multiplications [8].

We showed that tensor decomposition used for calculating sparse image representation via WTT is more effective way of image compression than using tensor decomposition directly.

WTT filters are signal-dependent, so we need to store it with compressed image. It significantly reduces efficiency of image compression via WTT because filters usually have high entropy.

We also showed that there is a possibility to make WTT filters signal-independent via tuning algorithm's parameters. When using pre-calculated signal-independent filter set, we have no need to save filters with encoded image. Efficiency of such filters is lower than efficiency of signal-dependent filters, but it is still better than saving filters with image. Comparison showed that efficiency of the introduced method is close to JPEG efficiency [8].

Moreover, WTT can be used for small blocks of image instead of applying WTT for the whole image. For example, we can use WTT for a block of size 8 by 8 by analogy of using discrete cosine transform (DCT) in JPEG. Our research showed that in this case we couldn't tune parameters of WTT to make filters signal-independent. It means that we need special filters for each block.

Also we analyzed efficiency of applying same filters to different blocks and saw that filters that were computed for a source block can be effective for blocks that are similar to the source block and also for completely different blocks [9]. Therefore, we can train dictionary of filters and use it to increase efficiency of the image compression.

Dictionary training includes finding filters that will be more effective than DCT and cover a lot of blocks. More precise training procedure was described in paper [9] where we also compared DCT to WTT with dictionary.

But we had an issue while using WTT with dictionary. How to choose right filters from dictionary for concrete input block? For this purpose we suggest three methods:

- The first and the simplest one is to use brute force method for choosing filters from the dictionary. This method is extremely slow and cannot be used in real time system for image compression. But this algorithm is still effective because it is guaranteed that the best filter for input block will be found if it is present in the dictionary. Also worth noting, that the speed of image reconstruction is still fast and close to JPEG image reconstruction speed.
- The second one is to use special hash functions. In this case we store additional hash values in the dictionary besides filters. When we handle input block we calculate hash for it and search in dictionary for the nearest hash value and choose filter which is associated with that value. This procedure reduces algorithm complexity of finding filters down to logarithm but that kind of search can lead to false positives especially in case of the large dictionary.
- Another method is to use machine learning techniques. This is a topic for further research. There is an assumption that the modern machine learning methods can help to find patterns between a specific set of filters and image blocks for which they can be effectively applied. If this is so, it will be possible to get rid of method's shortcomings and to reach speed comparable to the speed of the method number two (hash functions).

So, with using of WTT we can achieve efficiency of image compression similar to JPEG efficiency. Moreover, we can train dictionary in order to increase the efficiency of WTT.

The introduced method can be used as an alternative or as an extension to DCT-base compression algorithms. When it is used as an extension, it allows making existing algorithms of image and video compression more efficient.

Further ways of research in the area of using WTT for image compression:

- Detecting the optimal block size for WTT with a dictionary. We used blocks of 8 by 8 for ease of comparison with discrete cosine transform. But perhaps it would be better to use blocks of smaller size.
- Finding out the most efficient way of choosing filters from dictionary for applying it to input block of image.
- Using WTT for color image compression. There are a lot of ways to use tensor decomposition for color images.

REFERENCES

- [1] Dvorkovich V., Tchobanou M., Problems and prospects of the dynamic image coding systems, MediaVision, Moscow, 2011, vol. 2, pp. 55-64 (in Russian).
- [2] Bader B.W., Kolda T.G., Tensor decompositions and applications, SIAMRev, 2009, vol. 51, pp. 455-500.
- [3] Kiers H.A.L., A three-step algorithm for CANDECOMP/PARAFAC analysis of large data sets with multicollinearity, Journal of Chemometrics, 1998, vol. 12, pp. 171-255.
- [4] Terzopoulos D., Vasilescu M.A.O., Multilinear analysis of image ensembles: Tensor-Faces, 7th European Conference on Computer Vision, 2002, vol. 2350, pp. 447-460.
- [5] Oseledets I.V., Tensor-train decomposition, SIAM J. Sci. Comput, 2011, vol. 33, pp. 2295-2317.
- [6] Tchobanou M.K., Multidimensional multirate signal processing system, Moscow, Technosphere, 2009, 480 p. (in Russian).
- [7] Oseledets I.V., Tyrtysnikov E.E., Algebraic wavelet transform via quantics tensor train decomposition, INM RAS, 2010, pp. 1-19.
- [8] Tchobanou M.K., Makarov D.V. Image compression by using tensor approximation // Problems of Perspective Micro- and Nanoelectronic Systems Development - 2014. Proceedings / edited by A. Stempkovsky, Moscow, IPPM RAS, 2014. Part IV. P. 109-112 (in Russian).
- [9] Tchobanou M.K., Makarov D.V. Image compression via tensor approximations with pre-computed dictionary // Digital Signal Processing And Its Applications – DSPA 2015 / Moscow, Media Publisher, 2015. Volume 2. P. 711-715 (in Russian).
- [10] ISO/IEC 10918-1:1992 Informatontechnology – Digital Compression and Coding of Continuous-Tone Still Images – Requirements and Guidelines. ISO/IEC JTC1/SC29/WG10, 1992.